# A simple yet efficient algorithm for multiple kernel learning under elastic-net constraints

**Luca Citi** LCITI@IEEE.ORG
*School of Computer Science and Electronic Engineering*
*University of Essex*
*Colchester, CO4-3SQ, UK*

**Editor:** N/A

## 1. Introduction

This report presents an algorithm for the solution of multiple kernel learning (MKL) problems with elastic-net constraints on the kernel weights. Please see Sun et al. (2013) and Yang et al. (2011) for a review on multiple kernel learning and its extensions. In particular Yang et al. (2011) introduced the generalized multiple kernel learning (GMKL) model where the kernel weights are subject to elastic-net constraints.

While Xu et al. (2010) presents an elegant algorithm to solve MKL problems with $L_1$-norm and $L_p$-norm ($p \geq 1$) constraints, a similar algorithm is lacking in the case of MKL under elastic-net constraints. For example, algorithms based on the cutting plane method (Yang et al., 2011) require large and/or commercial libraries (*e.g.,* MOSEK).

The algorithm presented in this report provides an extremely simple and efficient solution to the elastic-net constrained MKL (GMKL) problem. Because it can be implemented in few lines of code and does not depend on external libraries (except a conventional $L_2$-norm SVM solver), it has a wider applicability and can be readily included in existing open-source machine learning libraries.

### 1.1 Notation

The symbol $\mathbb{R}_+^Q$ denotes the set of $Q$-dimensional vectors of nonnegative real numbers, while $\mathbb{R}_{++}^Q$ the set of vectors of strictly positive real numbers. The curled inequality symbols (*e.g.,* $\succ$) represent componentwise inequality. The symbol $\mathbf{1}_Q$ ($\mathbf{0}_Q$) denotes a $Q \times 1$ vector of all ones (zeros) while $\mathbf{e}_k$ is the vector with all entries zero except the $k$-th, which is one. The expression $a \circ b$ computes the componentwise product of the vectors $a$ and $b$. The notation $\theta_k$ refers to the $k$-th component of the vector $\theta$ while $\theta^{(m)}$ indicates the value of the vector $\theta$ at the $m$-th iteration of an iterative algorithm. For simplicity of notation, all summations involving $\ell$ go from 1 to $N$ (the number of training instances) while those involving $i$ or $k$ go from 1 to $Q$ (the number of kernels).

## 2. Elastic-net constrained MKL problem

### 2.1 Formulation of the generalized MKL problem

Given a set of labelled training data $\mathcal{D} = \{x_\ell, y_\ell\}_{\ell=1}^N$ where $x_\ell \in \mathcal{X}$ and $y_\ell \in \{-1, +1\}$, the learning problem corresponding to a generalized MKL classifier with elastic-net constraints (Yang et al., 2011) can be formulated as

$$\underset{\substack{\theta\in\Theta, b\in\mathbb{R},\\ \{f_k\in\mathcal{H}_k\}}}{\text{minimize}} \; \frac{1}{2}\sum_k \frac{\|f_k\|^2}{\theta_k} + C\sum_\ell L\Big(\sum_k f_k(x_\ell) - b, y_\ell\Big), \tag{1}$$

where $\mathcal{H}_k$ is the reproducing kernel Hilbert space (RKHS) associated with the $k$-th kernel, $L(\cdot)$ is the hinge loss function, and

$$\Theta = \{\theta \in \mathbb{R}_+^Q : \eta\|\theta\|_1 + (1-\eta)\|\theta\|_2^2 \leq 1\} \tag{2}$$

represents the elastic-net constraint on the kernel weights, with parameter $\eta \in [0, 1]$. When $\theta_k = 0$, $f_k$ must also be equal to zero (Rakotomamonjy et al., 2007) and the problem remains well-defined (under the convention $0/0 = 0$). Note that the minimization problem in (1) is a convex optimization problem because: a) the function to be minimized is jointly convex in its parameters $\theta$, $\{f_k\}$, and $b$ (Rakotomamonjy et al., 2007); and b) the search space is convex, in particular the elastic-net constraint $\Theta$.

### 2.2 Two-step block coordinate descent algorithm

The approach taken in this manuscript for the solution of (1) consists of a two-step block coordinate descent alternating between the optimization of the SVM classifiers and the optimization of the kernel weights. The procedure, which is reported in Algorithm 1, iterates until a stopping condition is met (see Section 2.3).

At iteration $m$, the first step minimizes problem (1) with respect to $\{f_k\}$ and $b$ for fixed values of the kernel weights $\theta^{(m)}$. As previously noted by others (Rakotomamonjy et al., 2007; Xu et al., 2010; Yang et al., 2011), this problem is equivalent to the standard SVM problem with a composite kernel $K(\cdot,\cdot) = \sum_k \theta_k^{(m)} K_k(\cdot,\cdot)$. Given the stack of Gram matrices $G_{k,\ell,\ell'}$, where $G_{k,\ell,\ell'} = K_k(x_\ell, x_{\ell'})$, existing SVM solvers can efficiently solve the composite SVM problem with Gram matrix $G^{(m)} = \sum_k \theta_k^{(m)} G_k$ and return the optimal bias $b^{(m)}$ and vector of dual coefficients $\alpha^{(m)}$.

The second step consists in minimizing (1) for $\theta \in \Theta$ while keeping $b$ and $\{f_k\}$ (or equivalently the dual coefficients) constant. Since the only term that depends on $\theta$ is the regularizer, we can define

$$\beta_k^{(m)} = \|f_k^{(m)}\|^2 = (\theta_k^{(m)})^2 \left[(\alpha^{(m)} \circ y)^\mathsf{T} G_k^{(m)} (\alpha^{(m)} \circ y)\right], \quad \forall k, \tag{3}$$

and attack this sub-problem as an instance of the more general problem of minimizing a weighted sum of reciprocals bound to elastic-net constraints:

$$\theta^{(m+1)} = \underset{\theta\in\Theta}{\arg\min} \sum_k \frac{\beta_k^{(m)}}{\theta_k}. \tag{4}$$

**Algorithm 1:** Solve elastic-net constrained MKL.

**Function** `SolveElNetMKL`$(G_1, \ldots, G_Q, y)$ **is**

```
// Initialization
θ ← 1_Q / s(1_Q);
for m ← 1 to maximum number of iterations do
    // Step 1: optimization of the SVM classifiers
    build composite Gram matrix: G ← Σ_k θ_k G_k;
    solve std SVM with G, y to get optimal dual coeffs and bias: α, b;
    // Step 1.5: check convergence
    for k ← 1 to Q do u_k ← [(α ∘ y)^T G_k (α ∘ y)];
    compute objective function (1) from dual form of SVM: Ō ← 1_Q^T α − u^T θ;
    solve elastic-net constrained LP: θ̆ ← SolveElNetLP(u) ;
    compute lower bound of (1): O̲ ← 1_Q^T α − u^T θ̆;
    if Ō/O̲ − 1 < ε_MKL then break;
    // Step 2: optimization of the kernel weights
    compute ‖f_k‖²: β ← θ ∘ θ ∘ u;
    solve elastic-net constr. weighted sum of recipr.: θ ← SolveElNetWSR(β, θ);
end
return θ, α, b;
```

**end**

Assuming positive definite kernels and excluding degenerate cases causing $\alpha^{(m)} = \mathbf{0}_Q$ (*e.g.,* all examples belonging to the same class), we have that $\beta^{(m)} \succ 0$ as long as $\theta^{(m)} \succ 0$. Because (4) diverges to $+\infty$ when any $\theta_k$ approaches zero, the minimization of (4) will always produce $\theta^{(m+1)} \succ 0$ as long as $\theta^{(m)} \succ 0$, *i.e.* ultimately provided that the initial point $\theta^{(0)} \succ 0$.

In the special case $\eta = 1$, the elastic-net constraint reduces to a lasso constraint and the problem (4) has a straightforward closed-form solution (Xu et al., 2010). In this manuscript, a novel, simple and efficient algorithm for the solution of this optimization problem in the general case $\eta \in [0, 1]$ is presented. Since the proposed solution to this sub-problem represents the novelty and main contribution of this paper, Section 3 will be entirely devoted to explaining this algorithm in detail.

### 2.3 Lower bound and stopping condition

Establishing a lower bound on the optimal value of the cost function (1) provides a non-heuristic stopping criterion for the two-step block coordinate descent algorithm. Following (Yang et al., 2011), the lower bound is found as the minimum over $\theta$ of the dual form of (1):

$$\underset{\theta \in \Theta}{\text{minimize}}\, \mathbf{1}_Q^\mathsf{T}\, \alpha - \frac{1}{2}\, (\alpha \circ y)^\mathsf{T} \left( \sum_k \theta_k G_k \right) (\alpha \circ y), \tag{5}$$

where $\alpha$ is the vector of dual coefficients of the composite SVM problem. In Yang *et al.* (2011), this bound is obtained as part of the cutting-plane method used for the optimization

of the kernel weights. The method proposed here takes a radically different approach as it finds the point $\check{\theta}$ where the minimum of (5) is attained as the solution of the elastic-net constrained linear program:

$$\underset{\theta \in \Theta}{\text{maximize}} \; u^{\mathsf{T}}\theta, \tag{6}$$

where

$$u_k = (\alpha \circ y)^{\mathsf{T}} G_k \, (\alpha \circ y), \quad \forall k. \tag{7}$$

A novel, simple and efficient algorithm for the solution of (6) is provided in Section 4.

At each iteration, problem (6) is solved for the current iterates $\alpha^{(m)}$ and $\theta^{(m)}$. The current value of the objective function and of the lower bound are simply computed as

$$\overline{\mathcal{O}}^{(m)} = \mathbf{1}_Q^{\mathsf{T}} \, \alpha^{(m)} - \frac{1}{2} \left(u^{(m)}\right)^{\mathsf{T}} \theta^{(m)} \qquad \underline{\mathcal{O}}^{(m)} = \mathbf{1}_Q^{\mathsf{T}} \, \alpha^{(m)} - \frac{1}{2} \left(u^{(m)}\right)^{\mathsf{T}} \check{\theta}^{(m)}. \tag{8}$$

The two-step block coordinate descent algorithm terminates when an iterate with relative gap $\overline{\mathcal{O}}^{(m)} / \underline{\mathcal{O}}^{(m)} - 1 < \epsilon_{\text{MKL}}$ is produced, which guarantees that the current value of the objective function $\overline{\mathcal{O}}^{(m)}$ is at most $\epsilon_{\text{MKL}} \, \mathcal{O}^{(\infty)}$ away from the optimal value $\mathcal{O}^{(\infty)}$.

## 3. Elastic-net constrained weighted sum of reciprocals

This whole section abstracts from the original MKL learning problem and focuses on the solution of the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_k \frac{\beta_k}{\theta_k} \\ \text{subj. to} \quad & \eta \|\theta\|_1 + (1-\eta)\|\theta\|_2^2 \leq 1, \\ & \theta \succeq 0 \end{aligned} \tag{9}$$

with $\beta \succ 0$ and $\eta \in [0, 1]$. As mentioned before, a solution to this problem must lie in the strictly positive orthant $\theta \succ 0$. Furthermore, the solution must be attained at a point where the elastic-net constraint is tight, *i.e.* $\eta \|\theta\|_1 + (1-\eta)\|\theta\|_2^2 = 1$. Aiming for a contradiction, let us assume that $\theta$ minimizes (9) with $\eta \|\theta\|_1 + (1-\eta)\|\theta\|_2^2 = 1 - \epsilon$, $0 < \epsilon \leq 1$. The point $\theta' = (1 + \frac{\epsilon}{3}) \, \theta$ clearly decreases the cost function while still satisfying the elastic-net constraint: $\eta \|\theta'\|_1 + (1-\eta)\|\theta'\|_2^2 < (1 + \frac{\epsilon}{3}) \eta \|\theta\|_1 + (1 + \frac{7}{9}\epsilon)(1-\eta)\|\theta\|_2^2 < (1+\epsilon)\,(1-\epsilon) < 1$. This contradicts the original assumption that $\theta$ was a minimum for (9). Therefore, we can search for the solution to (9) among the points in $\mathbb{R}_{++}^Q$ for which the elastic-net constraint holds with equality. Please notice that in the following of this section $x$ and $y$ simply denote vectors in $\mathbb{R}_{++}^Q$ (rather than training instances and labels like in the previous sections).

### 3.1 Re-scaled objective function

As a preliminary step in attacking the problem (9), we introduce an equivalent optimization problem. It is easy to verify that the norm

$$s(x) = \frac{\eta}{2} \|x\|_1 + \sqrt{\frac{\eta^2}{4} \|x\|_1^2 + (1-\eta) \|x\|_2^2} \tag{10}$$

verifies $\eta\left\|\frac{x}{s(x)}\right\|_1 + (1-\eta)\left\|\frac{x}{s(x)}\right\|_2^2 = 1$, $\forall x \in \mathbb{R}^Q \setminus \{0\}$. As a result, the change of variable $\theta = x/s(x)$, transforms the original problem (9) into the following equivalent one:

$$\underset{x \in \mathbb{R}^Q_{++}}{\text{minimize}} \quad h(x) = s(x)\, g(x) \tag{11}$$

where

$$g(x) = \sum_i \frac{\beta_i}{x_i}. \tag{12}$$

This new optimization problem implicitly accounts for the elastic-net constraint by means of the rescaling function $s$ which re-normalizes any $x \in \mathbb{R}^Q_{++}$ such that the vector $\theta = x/s(x)$ satisfies the elastic-net constraint with equality. Our new task is therefore to find a global minimum of $h$ in the positive orthant. Although $h$ is not a convex function, we can prove a weaker result — pseudoconvexity — which is still very useful in practice because critical points of pseudoconvex functions are also global minima (Cambini and Martein, 2008, theorem 3.2.5). In order to show that $h$ is pseudoconvex, the following theorem and its corollary are introduced (proofs in Appendix A.1).

**Theorem 1** *Let $A \subseteq \mathbb{R}^n$ be an open convex cone and $g, s : A \to \mathbb{R}_{++}$ be differentiable convex functions such that $s(cx) = cs(x)$ and $g(cx) = g(x)/c$ for all $c \in \mathbb{R}_{++}$ and $x \in A$. Their pointwise product $h(x) = s(x)\, g(x)$ is a pseudoconvex function in $A$.*

**Corollary 2** *Under the conditions of Theorem 1, all points $x = cx^*$ — with $c \in \mathbb{R}_{++}$ and $x^*$ satisfying $\nabla s(x^*) = -\nabla g(x^*)$ — are global minima for the function $h$, where it takes value $h(cx^*) = s^2(x^*) = g^2(x^*)$. If at least one of $s$ or $g$ is strictly convex, then $x^*$ is unique.*

The functions $s$ and $g$ defined in (12) and (10) satisfy the requirements for Theorem 1. In fact, they are both positive-valued differentiable functions in the positive orthant $\mathbb{R}^Q_{++}$ (which is an open convex cone) and they can be shown to be convex through some simple calculus. As a result of Theorem 1, $h$ is pseudoconvex function in $\mathbb{R}^Q_{++}$. Additionally, the strict convexity of $g$ guarantees the uniqueness of $x^*$ defined in Corollary 2.

### 3.2 Iterative minimization algorithm

The problem (11) can be minimized using the following novel iterative algorithm.[1] Given the current iterate $x^{(m)}$, the next iterate $x^{(m+1)}$ is generated as:

$$x_i^{(m+1)} = \sqrt{\frac{\beta_i}{q_i^{(m)}}} \tag{13a}$$

where

$$q_i^{(m)} = \nabla_i s(x^{(m)}) = \left.\frac{\mathrm{d}s(x)}{\mathrm{d}x_i}\right|_{x=x^{(m)}}. \tag{13b}$$

The algorithm is iterated until a stopping condition is met, at which point the last iterate $\hat{x}$ is re-scaled to obtain the solution to the problem (9) as $\hat{\theta} = \hat{x}/s(\hat{x})$. The pseudocode of the full algorithm — including the stopping condition that will be described in the following — is reported in Algorithm 2.

1. Please notice that the superscript $m$ now refers to the current iteration within the algorithm for the solution of (9) and is completely unrelated to the current iteration in the outer two-step block coordinate descent algorithm for the solution of the original MKL problem.

**Algorithm 2:** Solve elastic-net constrained weighted sum of reciprocals.

**Function** `SolveElNetWSR`($\beta$, $\theta^{(0)}$) **is**
  `// Initialization`
  $x \leftarrow \theta^{(0)}$;
  **for** $m \leftarrow 1$ **to** *maximum number of iterations* **do**
    `// Compute cost`
    $n_1 \leftarrow \sum_k x_k$;
    $r \leftarrow \sqrt{(\frac{\eta}{2})^2 n_1^2 + (1-\eta)\sum_k x_k^2}$;
    $s \leftarrow \frac{\eta}{2} n_1 + r$;
    $g \leftarrow \sum_k \beta_k / x_k$;
    `// Check convergence`
    **if** $m > 1$ **and** $s/g - 1 < \epsilon_{\text{wsr}}$ **then break**;
    `// Update iterate`
    $q \leftarrow \frac{\eta}{2} + [(\frac{\eta}{2})^2 n_1 + (1-\eta)\, x]/r$;
    **for** $k \leftarrow 1$ **to** $Q$ **do** $x_k \leftarrow \sqrt{\beta_k / q_k}$;
  **end**
  **return** $x/s$;
**end**

While a full proof of the convergence of the algorithm is provided in Section 3.3, the intuition behind it is sketched here. For ease of notation, we will hereafter drop the iteration superscript and refer to the current iterate as $w \triangleq x^{(m)}$ and to the next one as $z \triangleq x^{(m+1)}$. The new iterate $z$ generated from (13) can be interpreted as the solution to the problem:

$$\begin{aligned} \underset{x \in \mathbb{R}^Q_{++}}{\text{minimize}} \quad & \sum_i \frac{\beta_i}{x_i} \\ \text{subj. to} \quad & q^\mathsf{T} x = p, \end{aligned} \tag{14}$$

where $p = \sum_i \sqrt{\beta_i q_i}$. In other words, the new iterate is generated by minimizing the function $g$ on a hyperplane which is perpendicular to the gradient of $s$ at $w$. The specific choice of the offset constant, *i.e.* $p$ in (14), has an interesting geometrical interpretation. Because the functions $s$ and $g$ satisfy the requirements for Theorem 1, for any positive $c$ the point $y = cw$ is such that $s(y)g(y) = s(w)g(w)$ and also that $p = q^\mathsf{T} x = \nabla s(y)^\mathsf{T} x \leq s(x)\ \forall x$ (see Theorem 4 below). Choosing $c$ such that $s(y) = cs(w) = p$ and substituting (13) in (12), it is easy to show that the hyperplane $q^\mathsf{T} x = p$ has the following properties:

$$q = \nabla s(y) = -\nabla g(z), \tag{15a}$$
$$p = s(y) \leq s(x) \qquad \text{and} \tag{15b}$$
$$p = g(z) \leq g(x) \qquad \forall x : q^\mathsf{T} x = p. \tag{15c}$$

In other words, this hyperplane is externally tangent to the level sets of $s$ and $g$ of the same value, $p$. Asymptotically, the algorithm finds the hyperplane that is tangent to the two level sets at the same point. Although there is no guarantee that each step decreases both $g$ and

$s$, the next section will show that their product $h$ decreases monotonically at each step and that the algorithm, in fact, convergences towards the solution.

### 3.3 Convergence analysis

A fixed point for the iterative map (13) is the point $x^*$ satisfying the conditions of Corollary 2. In fact, substituting $q_i^* = \nabla_i s(x^*) = -\nabla_i g(x^*) = \beta_i/(x_i^*)^2$ in the iterate update (13a) makes it an identity. By Corollary 2, this fixed point is a global minimum for $h$ and, therefore, a solution for (11).

To show that the algorithm (13) can be used to solve (11), it remains to be proven that the iterative map (13) converges to its fixed point $x^*$ for all starting points $x^{(0)} \in \mathbb{R}^Q_{++}$. To do so, we will make use of convergence results of descent algorithms (Zangwill, 1969; Meyer, 1976; Bertsekas, 1999; Luenberger and Ye, 2008) and in particular of Zangwill's Global Convergence Theorem (Luenberger and Ye, 2008, p. 205), restated here for convenience.

**Theorem 3 (Global Convergence Theorem)** *Let $\mathcal{A}$ be an algorithm on $A$, and suppose that, given $x^{(0)}$, the sequence $\{x^{(m)}\}_{m=0}^{\infty}$ is generated satisfying $x^{(m+1)} \in \mathcal{A}(x^{(m)})$. Let a solution set $\Gamma \subset A$ be given, and suppose:*

1. *all points $x^{(m)}$ are contained in a compact set $S \subset A$,*
2. *there is a continuous function $\zeta$ on $A$ such that:*
   (a) *if $x \notin \Gamma$, then $\zeta(z) < \zeta(x)$ for all $z \in \mathcal{A}(x)$,*
   (b) *if $x \in \Gamma$, then $\zeta(z) \leq \zeta(x)$ for all $z \in \mathcal{A}(x)$,*
3. *the mapping $\mathcal{A}$ is closed at points outside $\Gamma$.*

*Then the limit of any convergent subsequence of $\{x^{(m)}\}$ is a solution.*

The following will show that Theorem 3 applies to the mapping $\{x^{(m+1)}\} = \mathcal{A}(x^{(m)})$ corresponding to (13). This mapping is defined in $A = \mathbb{R}^Q_{++}$ and has solution set $\Gamma = \{x^*\}$.

Since $s$ is a differentiable convex function in the open convex set $\mathbb{R}^Q_{++}$, it is actually continuously differentiable in $\mathbb{R}^Q_{++}$ (Rockafellar, 1970, Corollary 25.5.1). The specific choice of $s$ in (10) is such that $q_i^{(m)}$ is also strictly positive and, therefore, the iteration (13) defines a continuous function (point-to-point mapping) from $x^{(m)}$ to $x^{(m+1)}$. Since for a point-to-point mapping continuity implies closedness (Luenberger and Ye, 2008, p. 206), the third condition of Zangwill's theorem is satisfied.

As a first step towards verifying the second condition, the following theorem is introduced (proof provided in Appendix A.2).

**Theorem 4** *Given a norm $s : \mathbb{R}^n \to \mathbb{R}_+$ of the form $s(x) = d_0\|x\|_1 + \sqrt{d_1\|x\|_1^2 + d_2\|x\|_2^2}$ with $d_0, d_1, d_2 \geq 0$, the following property holds:*

$$\nabla s(y)^\mathsf{T} x \leq s(x) \leq \sqrt{x^\mathsf{T} \Lambda_y \, x} \quad \forall x, y \in \mathbb{R}^n, \tag{16}$$

*where $\Lambda_y$ is a diagonal matrix whose $i$-th diagonal element is $s(y)\nabla_i s(y)/y_i$.*

We can now write the following chain of inequalities showing that $h$ is non-increasing at each step:

$$h(x^{(m+1)}) \equiv h(z) \leq s^2(z) \leq z^\mathsf{T} \Lambda_y \, z = \sum_i \sqrt{\frac{\beta_i}{q_i}} \, s(y) \frac{q_i}{y_i} \sqrt{\frac{\beta_i}{q_i}} = h(y) = h(w) \equiv h(x^{(m)}), \tag{17}$$

where the first inequality follows from (15) while the second one from Theorem 4. Unfortunately, the fact that $h$ is constant along rays out of the origin makes it unsuitable as function $\zeta$ for Theorem 3 (the strict inequality in condition 2.(a) is violated for points $x = cx^*$ with $c \in \mathbb{R}_{++}$). Instead, we consider the function

$$\zeta(x) = 2\, h(x) + [s(x) - g(x)]^2 = g^2(x) + s^2(x), \tag{18}$$

for which the following inequality can be readily obtained from (15), (17), and (18):

$$\zeta(z) = g^2(z) + s^2(z) \leq 2\, s^2(z) \leq 2\, h(w) \leq \zeta(w). \tag{19}$$

Importantly, as prescribed by 2.(a), the expression (19) holds with equality only if the starting point of the iteration ($w$ in our case) is in the solution set $\Gamma$. This can be shown by first noticing that $\zeta(z) = \zeta(w)$ implies $g(z) = s(z) = g(w) = s(w)$. From the definition of $y$, we see that $s(w) = g(z) \;\Rightarrow\; y \equiv w$. Since the restriction of $g$ along $q^\mathsf{T} x = p$ is strictly convex, the inequality in (15c) holds as equality only at the minimum, *i.e.* $g(z) = g(y) \;\Rightarrow\; z \equiv y$. Putting these together, we obtain that $z \equiv w$, which substituted in (15a) finally yields $\nabla s(w) = -\nabla g(w)$, the condition defining the fixed point $x^*$. This proves that the second condition of Zangwill's theorem is also satisfied.

Through some simple algebra, it is easy to show that $\nabla_i s(x) \leq 1 \;\forall x, i$. This, together with (10) and (17), leads to $\sqrt{\beta_i} \leq x_i^{(m+1)} \leq s(x^{(m+1)}) \leq \sqrt{h(x^{(m)})} \leq \sqrt{h(x^{(0)})}$. As a result, all the points of the sequence (with the immaterial possible exception of $x^{(0)}$) are contained in $[\min_i \sqrt{\beta_i}, \sqrt{h(x^{(0)})}]^Q$, which is a closed and bounded subset of $\mathbb{R}^Q_{++}$, as prescribed by the first condition of the theorem.

In conclusion, we have proven that the algorithm defined by the iteration (13) satisfies the conditions of Zangwill's theorem. Also, because the solution set $\Gamma$ consists of a single point $x^*$, the sequence $\{x^{(m)}\}$ converges to $x^*$ (Luenberger and Ye, 2008, p. 206).

### 3.4 Stopping condition

We now establish a lower bound on the optimal value of $h$, which will be used to provide a non-heuristic stopping criterion for the iterative algorithm in (13). Given the solution $x^*$ and the new iterate $x^{(m+1)}$ obtained as described in Section 3.2, we observe that, since $q$ and $x^*$ lie in the (strictly) positive orthant, there always exists $c \in \mathbb{R}_{++}$ such a that $q^\mathsf{T}(cx^*) = p$, with $p$ as in Section 3.2. Therefore, (15c) implies $p \leq g(cx^*)$ and Theorem 4 yields $p = q^\mathsf{T}(cx^*) \leq s(cx^*)$. Combining these two inequalities gives $p^2 \leq g(cx^*)\, s(cx^*)$, which can be rewritten as $g^2(x^{(m+1)}) \leq h(x^*)$ where the equality only holds at the solution $x^*$. As a result, $h(x^{(m+1)}) - g^2(x^{(m+1)})$ bounds how suboptimal the iterate is, even without knowing the exact value of $h(x^*)$. The following stopping condition guarantees a predefined relative accuracy $\epsilon_{\text{wsr}} > 0$:

$$\frac{h(x^{(m+1)}) - g^2(x^{(m+1)})}{g^2(x^{(m+1)})} = \frac{s(x^{(m+1)})}{g(x^{(m+1)})} - 1 \leq \epsilon_{\text{wsr}}. \tag{20}$$

The algorithm terminates after an $\epsilon_{\text{wsr}}$-suboptimal iterate is produced, *i.e.* when (20) is satisfied, which guarantees that $h(x^{(m+1)}) - h(x^*) \leq \epsilon_{\text{wsr}}\, h(x^*)$.

### 3.5 Alternative approaches

This section presents a brief overview of alternative approaches that were devised by the author in the process of creating and improving the main method presented above. They are reported here because they may be advantageous in specific situations and for some values of the parameters.

An approach to minimizing (9), which works particularly well when $\eta$ is small, is by using the alternative update:

$$x_i^{(m+1)} = \left( \frac{x_i^{(m)} \beta_i}{q_i^{(m)}} \right)^{\frac{1}{3}} \tag{21}$$

instead of (13a). For an appropriate choice of $p' > 0$, this iterate is the solution to the problem:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^Q_{++}}{\text{minimize}} & \displaystyle\sum_i \frac{\beta_i}{x_i} \\ \text{subj. to} & x^{\mathsf{T}} \Lambda_{x^{(m)}}\, x = p', \end{array} \tag{22}$$

which is an analogous of (14) using a quadratic constraint instead of a linear one. The iterative map defined by (21) has the same fixed point as the map (13a) and a convergence proof can be obtained using arguments similar to those in Section 3.3. In simulations, the convergence rate of the update rule (21), appears to be marginally better than (13a) for small values of $\eta$ (less than approximately 0.25) and significantly worse otherwise. For this reason, it may be advantageous to use (13a) when $\eta \geq 0.25$ and alternate between (21) and (13a) when $\eta < 0.25$.

An algorithm for the solution of the problem (9) using a majorization-minimization (MM) procedure was presented in (Citi, 2015). Briefly, the algorithm is similar to coordinate descent but at each step — instead of performing a full line search to minimize $h$ as a function of one of the optimization variables — it reduces it by minimizing a carefully designed surrogate function, called a majorizer, which can be solved in closed form. The number of iterations required to obtain a given accuracy is comparable to that of Algorithm 2 but each iteration requires roughly four times as many flops.

## 4. Elastic-net constrained linear program

This section introduces an efficient algorithm for finding the solution $\check{\theta}$ of the elastic-net constrained linear program:

$$\begin{array}{ll} \text{maximize} & u^{\mathsf{T}} \theta \\ \text{subj. to} & \eta \|\theta\|_1 + (1-\eta)\|\theta\|_2^2 \leq 1, \\ & \theta \succeq 0 \end{array} \tag{23}$$

with $u \succeq 0$, $u \neq \mathbf{0}_Q$ and $\eta \in [0, 1]$. As shown in Section 2.3, a solution to this problem provides a lower bound on the optimal value of the original MKL cost function (1).

### 4.1 Algorithm

In the special case $\eta = 1$, the (possibly nonunique) straightforward solution to the problem is the vector $\mathbf{e}_k$, where $k$ is such that $u_k = \max_i u_i$. When $\eta < 1$, simple algebra shows that points in $\mathbb{R}_+^Q$ satisfy the elastic-net constraint if and only if they also belong to the hyper-sphere with centre $c$ and radius $r$, where

$$d = \eta/(2 - 2\eta), \tag{24}$$

$$c = -d\,\mathbf{1}_Q, \tag{25}$$

$$r = \sqrt{Q\,d^2 + 2d + 1}. \tag{26}$$

Therefore, the problem (23) is equivalent to:

$$\underset{\substack{\theta \in \mathbb{R}_+^Q, \\ \|\theta - c\|_2^2 \leq r^2}}{\text{maximize}}\ u^\mathsf{T}\theta. \tag{27}$$

Let us now consider the point $q$:

$$q = r\,u/\|u\|_2 + c, \tag{28}$$

which is the point of the hyper-sphere which is farthest away in the direction of $u$. If this point is also in $\mathbb{R}_+^Q$, then $\check{\theta} = q$ is trivially a solution for the optimization problem (27). If this is not the case, the important property that $q_k < 0 \;\Rightarrow\; \check{\theta}_k = 0$ (of which a proof is provided in Section 4.2) suggests a method to incrementally prune away coordinate directions that are guaranteed to be zero in the optimal solution $\check{\theta}$. At each iteration $m$, the algorithm keeps track of the set $Z^{(m)}$ of indices for which it has already been established that the corresponding element of $\check{\theta}$ is null, *i.e.* $k \in Z^{(m)} \Rightarrow \check{\theta}_k = 0$. The set $Z$ is initialized to the empty set $\emptyset$ at the beginning of the algorithm and grows monotonically at each iteration. We denote as $|Z|$ the cardinality of $Z$, as $\overline{Z}$ its complement and as $u_{\overline{Z}}$ the projection of $u$ on the $(Q-|Z|)$-dimensional subspace spanned by coordinate directions corresponding to indices in $\overline{Z}$. The algorithm generates the next iterate $q^{(m)}$ according to:

$$q_k^{(m)} = \begin{cases} r^{(m)}\,u_k/\|u_{\overline{Z}^{(m)}}\|_2 - d, & \text{if } k \in \overline{Z}, \\ 0 & \text{if } k \in Z. \end{cases} \tag{29}$$

This is the point of the $|\overline{Z}^{(m)}|$-dimensional disc of radius $r^{(m)} = \sqrt{|\overline{Z}^{(m)}|\,d^2 + 2d + 1}$ and centre $c_{\overline{Z}^{(m)}}$ which is farthest away in the direction of $u$. If any of the elements of $q^{(m)}$ is negative, their indices are added to $Z$ and the algorithm starts a new iteration, otherwise the algorithm ends and the last iterate is returned as the solution $\check{\theta}$ to the elastic-net constrained linear program (23). The detailed algorithm is reported in Algorithm 3.

### 4.2 Convergence analysis

The fact that the greedy algorithm presented in Section 4.1 finds the global solution in a finite number of iterations stems from the property that if the algorithm produces an iterate with a negative component, the corresponding element of the solution must be zero:

$$q_k^{(m)} < 0 \;\Rightarrow\; \check{\theta}_k = 0, \quad \forall\, k, m. \tag{30}$$

**Algorithm 3:** Solve elastic-net constrained linear program.

**Function** `SolveElNetLP(`$u$`)` **is**

```
// Initialization
```
$Z \leftarrow \emptyset$;
$d \leftarrow \eta/(2-2\eta)$;
**do**
```
// Main loop
```
$r \leftarrow \sqrt{|\overline{Z}|\,d^2 + 2d + 1}$;
$q_{\overline{Z}} \leftarrow r\,u_{\overline{Z}}/\|u_{\overline{Z}}\|_2 - d$;
$N \leftarrow \{k \,|\, q_k < 0\}$;
$q_N \leftarrow 0$;
$Z \leftarrow Z \cup N$;

**while** $N \neq \emptyset$;
**return** $q$;

**end**

Aiming for a contradiction, let us assume that $\theta$, with $\theta_k > 0$, is a solution to (23) and that at some point the algorithm produces the iterate $q^{(m)}$ with $q_k^{(m)} < 0$. For conciseness, we denote $q^{(m)}$ simply as $q$, $Z^{(m)}$ as $Z$, $r^{(m)}$ as $\rho$, and $u_{\overline{Z}^{(m)}}$ as $w$, within this section. From (29), it follows that $q_k < 0$ implies $w_k < \|w\|\,d/\rho$. Let us consider the point

$$\theta' = \theta - \epsilon\,\mathbf{e}_k + \delta\,w, \quad \text{with } 0 < \epsilon \leq \theta_k \text{ and } \delta = \frac{d\,\epsilon}{\rho\,\|w\|}, \tag{31}$$

and show that it satisfies the constraints of (27). Because $\epsilon \leq \theta_k$, $\delta > 0$, and $w \succeq 0$, then $\theta \succeq 0 \;\Rightarrow\; \theta' \succeq 0$. It is now sufficient to show that $\|\theta - c\|^2 \leq r \;\Rightarrow\; \|\theta' - c\|^2 \leq r$:

$$\begin{aligned}
\|\theta' - c\|^2 &= \|\theta - c\|^2 + \|\delta\,w - \epsilon\,\mathbf{e}_k\|^2 + 2\,(\delta\,w - \epsilon\,\mathbf{e}_k)^\mathsf{T}(\theta - c) \\
&= \|\theta - c\|^2 + \frac{d^2\epsilon^2}{\rho^2} + \epsilon^2 - 2\,\delta\,\epsilon\,w_k + 2\,\frac{d\,\epsilon}{\rho\,\|w\|}\,w^\mathsf{T}(\theta_{\overline{Z}} - c_{\overline{Z}}) - 2\,\epsilon\,\theta_k - 2\,d\,\epsilon \\
&\leq \|\theta - c\|^2 + (2\,\epsilon^2 - 2\,\epsilon\,\theta_k) - 2\,\delta\,\epsilon\,w_k + 2\,d\,\epsilon\,\left(\frac{\|w\|\,\|\theta_{\overline{Z}} - c_{\overline{Z}}\|}{\|w\|\,\rho} - 1\right) \\
&\leq \|\theta - c\|^2 + 2\,d\,\epsilon\,\left(\frac{\sqrt{\|\theta - c\|^2 - |Z|\,d^2}}{\rho} - 1\right) \leq \|\theta - c\|^2
\end{aligned} \tag{32}$$

This proves that $\theta'$ is a feasible point for (27). Because $u^\mathsf{T}\theta' = u^\mathsf{T}\theta - \epsilon\,u_k + \delta\,u^\mathsf{T}w = u^\mathsf{T}\theta - \epsilon\,w_k + \frac{d\,\epsilon}{\rho\,\|w\|}\|w\|^2 > u^\mathsf{T}\theta$, the feasible point $\theta'$ improves over $\theta$, which therefore cannot be a solution. This contradiction proves (30).

## 5. Conclusions

This technical report focuses on an algorithm for the minimization of a positive-weighted sum of reciprocals bound to elastic-net constraints. This algorithm, explained in detail

in Sections 3.1–3.4, can be used to optimize the kernel weights within a two-step block coordinate descent alternating between the optimization of the SVM classifiers and the optimization of the kernel weights. Preliminary tests (not reported) of the computational cost of the algorithm show that it compares very favourably to existing and alternative approaches. Finally, because it does not depend on external libraries, it has a wide applicability and can be readily included in existing open-source machine learning libraries.

## Appendix A. Proofs of theorems

The proofs of the theorems given in the text are reported in this appendix in the form of structured proofs as advocated by Leslie Lamport (2012). Each assertion follows from previously stated facts, which are explicitly named to tell the reader exactly which ones are being used at each step.

### A.1 Proofs of Theorem 1 and Corollary 2

**Theorem 1** *Let $A \subseteq \mathbb{R}^n$ be an open convex cone and $g, s : A \to \mathbb{R}_{++}$ be differentiable convex functions such that $s(cx) = cs(x)$ and $g(cx) = g(x)/c$ for all $c \in \mathbb{R}_{++}$ and $x \in A$. Their pointwise product $h(x) = s(x)\,g(x)$ is a pseudoconvex function in $A$.*

**Proof**

1. To show that the differentiable function $h : A \to \mathbb{R}_{++}$ defined in an open convex set is pseudoconvex, it suffices to assume for the remaining of this proof that:

   1.1. $y, z \in A$,

   1.2. $h(z) < h(y)$,

   and prove that $\nabla h(y)^\mathsf{T}(z - y) < 0$.

   Proof: By the definition of pseudoconvex function (Cambini and Martein, 2008, definition 3.2.1).

2. $\forall\, x \in A : \nabla g(x)^\mathsf{T} x = -g(x)$.

   Proof: By differentiating $g(cx) = g(x)/c$ w.r.t. $c$ and evaluating it for $c = 1$.

3. $\forall\, x \in A : \nabla s(x)^\mathsf{T} x = s(x)$.

   Proof: By differentiating $s(cx) = cs(x)$ w.r.t. $c$ and evaluating it for $c = 1$.

4. $\forall\, x \in A : \nabla h(x)^\mathsf{T} x = g(x)\nabla s(x)^\mathsf{T} x + s(x)\nabla g(x)^\mathsf{T} x = 0$.

   Proof: Follows directly from 2 and 3.

5. Given $z$ and $y$ as in 1.1, $\exists\, c \in \mathbb{R}_{++}$ such that the point $z' = cz$ satisfies $h(z') = h(z)$ and $s(z') = s(y)$.

   Proof: For any positive $c$ the corresponding $z'$ is in $A$ (because $A$ is a cone) and satisfies the first condition: $h(z') = s(cz)\,g(cz) = cs(z)\,g(z)/c = h(z)$. We choose $c = s(y)/s(z)$ which also satisfies the second condition: $s(z') = s(y)/s(z)\,s(z) = s(y)$.

6. $\forall\, x, x' \in A : \nabla s(x)^{\mathsf{T}} x' \leq s(x')$.

   Proof: The first-order conditions for convexity (Boyd and Vandenberghe, 2009, ch 3.1.3) imply $s(x') \geq s(x) + \nabla s(x)^{\mathsf{T}}(x' - x)$. Substituting 3 and rearranging yields 6.

7. $\forall\, x, x' \in A : \nabla g(x)^{\mathsf{T}} x' \leq g(x') - 2\, g(x)$.

   Proof: The first-order conditions for convexity imply $g(x') \geq g(x) + \nabla g(x)^{\mathsf{T}}(x' - x)$. Substituting 2 and rearranging yields 7.

8. $\nabla h(y)^{\mathsf{T}} z' < 0$.

   Proof: By 6, 7, 5 and 1.2, we have:

$$\begin{aligned} \nabla h(y)^{\mathsf{T}} z' &= g(y)\, \nabla s(y)^{\mathsf{T}} z' + s(y)\, \nabla g(y)^{\mathsf{T}} z' \\ &\leq g(y)\, s(z') + [s(y)\, g(z') - 2\, s(y)\, g(y)] \\ &= h(y) + h(z') - 2\, h(y) \\ &= h(z') - h(y) = h(z) - h(y) < 0. \end{aligned}$$

9. Q.E.D.

   Proof: By 8, 5 and 4, we have:

$$c\nabla h(y)^{\mathsf{T}} z < 0 \;\Rightarrow\; \nabla h(y)^{\mathsf{T}} z = \nabla h(y)^{\mathsf{T}}(z - y) < 0.$$

   By 1, the latter proves the theorem.

■

**Corollary 2** *Under the conditions of Theorem 1, all points $x = cx^*$ — with $c \in \mathbb{R}_{++}$ and $x^*$ satisfying $\nabla s(x^*) = -\nabla g(x^*)$ — are global minima for the function $h$, where it takes value $h(cx^*) = s^2(x^*) = g^2(x^*)$. If at least one of $s$ or $g$ is strictly convex, then $x^*$ is unique.*

**Proof**

10. $\nabla s(x^*) = -\nabla g(x^*) \;\Rightarrow\; s(x^*) = g(x^*)$.

    Proof: Follows immediately from the statements 2 and 3 of the proof of Theorem 1.

11. $x^*$ is a critical point for $h$.

    Proof: From the condition $\nabla s(x^*) = -\nabla g(x^*)$ and from statement 10: $\nabla h(x^*) = g(x^*)\, \nabla s(x^*) + s(x^*)\, \nabla g(x^*) = 0$.

12. $x^*$ is a global minimum of $h$.

    Proof: Because $x^*$ is a critical point (statement 12) of a pseudoconvex function (Theorem 1), it is also a global minimum (Cambini and Martein, 2008, theorem 3.2.5).

13. If at least one of $s$ or $g$ is strictly convex, then $x^*$ is unique.

    Proof: Aiming for a contradiction, let us assume that there is a point $\hat{x} \in A \backslash \{x^*\}$ such that $\nabla s(\hat{x}) = -\nabla g(\hat{x})$. By using the same reasoning as in 10, this implies $s(\hat{x}) = g(\hat{x})$. Without loss of generality, let us assume that $s(x^*) \geq s(\hat{x})$ and that $s$ is strictly convex. From the first-order conditions for (strict) convexity, we obtain:

$$s(\hat{x}) > s(x^*) + \nabla s(x^*)^\mathsf{T}(\hat{x} - x^*) \quad \Rightarrow \quad \nabla s(x^*)^\mathsf{T}(\hat{x} - x^*) < 0, \tag{33}$$

$$g(\hat{x}) \geq g(x^*) + \nabla g(x^*)^\mathsf{T}(\hat{x} - x^*) \quad \Rightarrow \quad \nabla s(x^*)^\mathsf{T}(\hat{x} - x^*) \geq 0, \tag{34}$$

    which is obviously a contradiction.

14. Q.E.D.

    Proof: From 10, 12, 13, and the definitions of $s$, $g$, and $h$ in the statement of Theorem 1.

■

### A.2 Proof of Theorem 4

**Theorem 4** *Given a norm $s : \mathbb{R}^n \to \mathbb{R}_+$ of the form $s(x) = d_0\|x\|_1 + \sqrt{d_1\|x\|_1^2 + d_2\|x\|_2^2}$ with $d_0, d_1, d_2 \geq 0$, the following property holds:*

$$\nabla s(y)^\mathsf{T} x \leq s(x) \leq \sqrt{x^\mathsf{T} \Lambda_y \, x} \quad \forall x, y \in \mathbb{R}^n, \tag{16}$$

*where $\Lambda_y$ is a diagonal matrix whose $i$-th diagonal element is $s(y)\nabla_i s(y)/y_i$.*

**Proof**

1. $\forall\, y \in \mathbb{R}^n : \nabla s(y)^\mathsf{T} y = s(y)$.

   Proof: Since $s$ is a norm, $s(cy) = |c|s(y)$. By differentiating both sides w.r.t. $c$ and evaluating it for $c = 1$, we obtain the statement 1.

2. $\forall\, y, x \in \mathbb{R}^n : \nabla s(y)^\mathsf{T} x \leq s(x)$.

   Proof: The first-order conditions for convexity (Boyd and Vandenberghe, 2009, ch 3.1.3) imply $s(x) \geq s(y) + \nabla s(y)^\mathsf{T}(x - y)$. Substituting 1 and rearranging yields 2.

3. Define $r : \mathbb{R}^n \to \mathbb{R}_+$ as $r(y) = \sqrt{d_1^2\|y\|_1^2 + d_2^2\|y\|_2^2}$.

4. $\forall\, y, x \in \mathbb{R}^n : \sqrt{\dfrac{d_1^2\|x\|_1^2 + d_2^2\|x\|_2^2}{\|x\|_1^2}} \leq \dfrac{1}{2}\left[\dfrac{r(y)}{\|y\|_1} + \dfrac{d_1^2\|x\|_1^2 + d_2^2\|x\|_2^2}{\|x\|_1^2}\dfrac{\|y\|_1}{r(y)}\right]$.

   Proof: From the inequality $\sqrt{z} \leq \frac{1}{2}[\sqrt{z_0} + z/\sqrt{z_0}]$, which in turn results from the concavity of the square root function.

5. $\forall\, y, x \in \mathbb{R}^n : s^2(x) \leq s(y)\left[\dfrac{d_0}{\|y\|_1}\|x\|_1^2 + \dfrac{d_1}{r(y)}\|x\|_1^2 + \dfrac{d_2}{r(y)}\|x\|_2^2\right]$.

   Proof: Follows from writing out the lhs explicitly using the definition of $s$ and then exploiting the statement in 4.

6. $\forall\, y, x \in \mathbb{R}^n : \frac{d_0}{\|y\|_1}\|x\|_1^2 + \frac{d_1}{r(y)}\|x\|_1^2 + \frac{d_2}{r(y)}\|x\|_2^2 \leq \sum_i \frac{d_0}{|y_i|}x_i^2 + \sum_i \frac{d_1\|y\|_1}{r(y)|y_i|}x_i^2 + \sum_i \frac{d_2}{r(y)}x_i^2$.

   Proof: The last term of each side of the inequality is identical. Applying Radon's inequality it is easy to show that the each one of the first two terms of the lhs is bounded by the corresponding term in the rhs.

7. $\forall\, y, x \in \mathbb{R}^n : s^2(x) \leq x^\mathsf{T}\Lambda_y\, x$.

   Proof: Follows from combining 5 and 6, then using the definition of $\Lambda_y$.

8. Q.E.D.

   Proof: Combining 2 and 7 proves the theorem.

■

## References


Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Mass, 2nd edition edition, September 1999. ISBN 978-1-886-52900-7.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009. ISBN 978-0-521-83378-3.

Alberto Cambini and Laura Martein. *Generalized convexity and optimization: Theory and applications*, volume 616. Springer, 2008. ISBN 978-3-540-70875-9.

Luca Citi. Elastic-net constrained multiple kernel learning using a majorization-minimization approach. In *Proceedings of the 7th Computer Science and Electronic Engineering Conference (CEEC)*, pages 29–34, 2015. doi: 10.1109/CEEC.2015.7332695.

Leslie Lamport. How to write a 21st century proof. *Journal of Fixed Point Theory and Applications*, 11(1):43–63, 2012. doi: 10.1007/s11784-012-0071-6.

David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer Science & Business Media, June 2008. ISBN 978-0-387-74503-9.

Robert R. Meyer. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *Journal of Computer and System Sciences*, 12(1):108–121, February 1976. doi: 10.1016/S0022-0000(76)80021-9.

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning*, pages 775–782. ACM, 2007.

Ralph T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. ISBN 0-691-08069-0.

Tao Sun, Licheng Jiao, Fang Liu, Shuang Wang, and Jie Feng. Selective multiple kernel learning for classification with ensemble strategy. *Pattern Recognition*, 46(11):3081–3090, November 2013. doi: 10.1016/j.patcog.2013.04.003.

Zenglin Xu, Rong Jin, Haiqin Yang, Irwin King, and Michael R. Lyu. Simple and efficient multiple kernel learning by group lasso. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1175–1182, 2010.

Haiqin Yang, Zenglin Xu, Jieping Ye, I King, and M R Lyu. Efficient sparse generalized multiple kernel learning. *IEEE Transactions on Neural Networks*, 22(3):433–446, March 2011. doi: 10.1109/TNN.2010.2103571.

Willard I. Zangwill. *Nonlinear programming: a unified approach*. Prentice-Hall, 1969. ISBN 978-0-136-23579-8.